

LEAP COPTER

Steuerung einer virtuellen Drone mit Leap Motion

*Stefan Berner, Aleksandar Presic
WS 2015/16*

*Interaction Engineering
Prof. Dr. Michael Kipp
Augsburg University of Applied Sciences*



Hochschule
Augsburg University of
Applied Sciences

LEAP COPTER

Steuerung einer virtuellen Drone mit Leap Motion

ABSTRACT ENGLISH

LeapCopter is a simulation with intuitive handgestures for quadcopters. Conventional controlling becomes redundant with the Leap Motion sensor, because the aircraft is controlled with various handgestures. Our simulation provides the pilot the possibility to get to know the controls in a safe environment, therefore the real copter isn't exposed to any threats at any time. Afterwards he can steer his real copter with his newly learned Knowledge. Two implemented one-hand controls provide as a base for comparison.

ABSTRACT

LeapCopter ist eine Simulation eines Quadcopters mit intuitiven Handgesten. Mit dem Leap Motion Sensors wird eine herkömmliche Steuerung überflüssig, da das Fluggerät nur mit verschiedenen Handbewegungen gesteuert wird. Unsere Simulation bietet dem Benutzer die Möglichkeit die Steuerung kennen zu lernen, um

anschließend das Gelernte mit einer echten Drone einsetzen zu können. Während der Eingewöhnungsphase im Simulator kann risikofrei geflogen werden und das teure Fluggerät wird keinen Gefahren ausgesetzt. Mit zwei implementierten Ein-Hand-Steuerungen kann ein direkter Vergleich gezogen und die Steuerung gewählt werden, mit die einem besser gefällt.

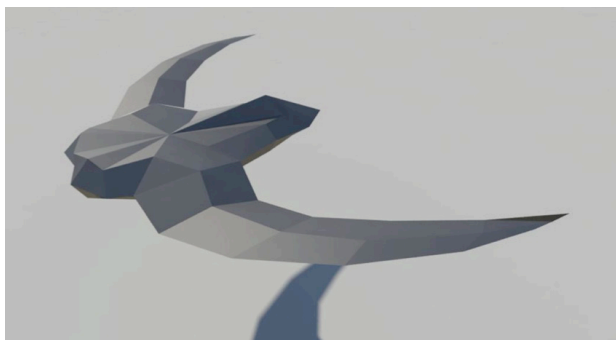
EINLEITUNG

Die Benutzung von Dronen erfreut sich einer immer größerer Beliebtheit, und damit auch die Entwicklung von alternativen Steuerungsmöglichkeiten, um die konventionellen und zum Teil unhandlichen Fernbedienungen zu ersetzen. Neben der Möglichkeit das Gerät mit Hilfe des Smartphones zu steuern, kann die Bedienung auch mit reinen Handbewegungen durchgeführt werden. Der Leap Motion Sensor ermöglicht das Erkennen von Gesten die anschließend auf das Fluggerät übertragen werden können. Allerdings möchten die

wenigsten eine solche Steuerung mit einem echten Quadcopter testen, da das Risiko zu hoch ist das teure Gerät zu beschädigen. In der Simulation kann der Benutzer sich sicher und einfach eingewöhnen um später die Drone in keine Gefahren auszusetzen. Der erkannten Problematik, möchten wir mit einer implementierten Simulation entgegen wirken.

REFERENZ

Eine verwandte Arbeit ist die Bachelor Arbeit von Sebastian Huber, der sich mit einer Gestensteuerung von Flugsimulatoren befasst hat. Der Ansatz bezieht sich allerdings ebenfalls nicht auf eine Anwendung mit echten Fluggeräten, sondern nur mit virtuellen Simulationen. Um die Gestensteuerung zu ermöglichen greift er ebenfalls auf den Leap Motion Sensor zurück. Der Sensor erkennt mit den verbauten Infrarotkameras die Position und Neigung von Hand und Fingern genauestens. In seiner Arbeit deckt er nicht nur die Gesten für die Steuerung der Flugrichtungen ab sondern auch die zusätzlichen Funktionen wie z.B das Schießen, Zielen und Bodenbremsen.



1. <http://michaelkipp.de/student/Huber2014.pdf>/ Seite 1

ERSTE AUSEINANDERSETZUNG

Für unsere Arbeit sind die Handgesten das Kernelement, um eine leichte und verständliche Steuerung entwickeln zu können. Die Gestensteuerung bei Dronen relativ neu und undefiniert ist, dadurch müssen alle Gesten durchdacht und auf ihre Sinnhaftigkeit überprüft werden. Die erste Auseinandersetzung basiert auf dem Vergleich einer Zweihandsteuerung und einer Einhandsteuerung. Dabei konzentrieren wir uns auf das reine Steuern des Fluggeräts. Zusätzliche Funktionen, wie Sie bei Quadcoptern mit Videokameras vorhanden sind, werden hierzu nicht berücksichtigt. Wir beschränken uns deshalb auf die Kernelemente der Steuerung für Vorwärts / Rückwärts / Links / Rechts / Hoch / Runter. Die ersten Ansätze der Zweihandsteuerung offenbarte mehrere Problemen. Die wichtigste Erkenntnis war, dass es sehr schwierig ist, gleichzeitig verschiedene Gesten mit den jeweiligen Händen auszuführen. Der Benutzer muss bei jedem Befehl umdenken, was zu einer größeren Komplexität beim Fliegen führt. Dadurch ist die Zweihandsteuerung für unser Projekt irrelevant, da wir uns auf eine einfache und verständliche Bedienung fokussieren möchten. Daraus ergaben sich zwei verschiedene Einhandsteuerungen, die beide in unserem Prototyp implementiert wurden.

VARIANTE A:

Steuerung per Neigung

Die erste Variante berücksichtigt den Winkel der Handfläche. Die Ausrichtung und Winkel definieren so-

wohl die Flugrichtung als auch die Geschwindigkeit. Je größer der Neigungswinkel der Handfläche, desto höher die Geschwindigkeit.

VORWÄRTS UND RÜCKWÄRTS:

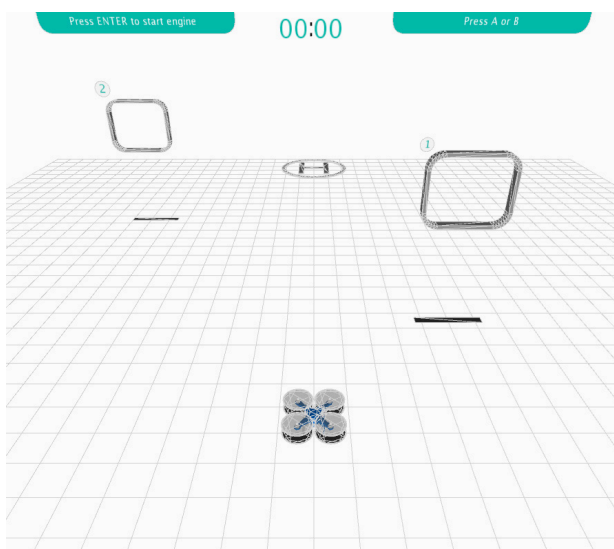
Die Hand steht über dem Sensor in der Ausgangsposition, durch das Senken der Handfläche nach Vorne um ca. 15 Grad, bewegt sich die Drone nach Vorne. Um das Gerät nach Hinten zu bewegen wird die Handfläche nach Oben um ca. 15 Grad aufgestellt.

LINKS UND RECHTS:

Die Hand befindet sich wieder flach in der Ausgangsposition. Um nach Links bzw. Rechts zu fliegen, wird die Handfläche um bis zu ca. 60 Grad nach links bzw. rechts gedreht.

STEIGEN UND SINKEN:

Beim Starten befindet sich die Hand in der Ausgangsposition, die sich ca. 15cm über dem Sensor befindet.



2. Abbildung der Anwendung

det. Befindet sich die Hand über der Ausgangsposition, steigt die Drone in die Höhe. Ist die Hand unter der Ausgangsposition, geht der Copter in den Sinkflug über. Die Geschwindigkeit wird über den Abstand zwischen Ausgangsposition und aktueller Handposition geregelt, je größer der Abstand, desto höher die Geschwindigkeit.

VARIANTE B:

Steuerung per Entfernung

Die zweite Variante berücksichtigt die Entfernung der Handfläche zum Ausgangspunkt. Je weiter sich die Hand vom Ausgangspunkt befindet, desto höher die Geschwindigkeit.

VORWÄRTS UND RÜCKWÄRTS:

Die Vor- und Rückwärts-Steuerung wird durch das Bewegen der flachen Hand nach Vorne bzw. nach Hinten über den Sensor hinaus ausgeführt. Hierzu sind Bewegungen von bis 30cm möglich.

LINKS UND RECHTS:

Um die Drone Links bzw. Rechts zu steuern, wird die Hand nach Links bzw. Rechts bewegt. Die Geschwindigkeit wird ebenfalls über den Abstand beeinflusst und auch hier kann die Hand bis zu 30cm vom Ausgangspunkt bewegt werden.

STEIGEN UND SINKEN:

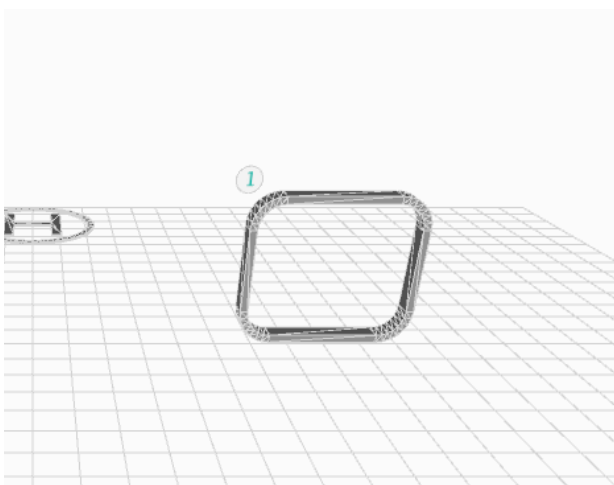
Die Steuerung der Höhe ist bei beiden Varianten identisch, da das Verändern der Höhe mit Heben oder Senken der Hand am natürlichsten ist und sich daher für beide Varianten eignet.

PARKOUR:

Um die beiden Varianten zu vergleichen wird ein Parkour aufgesetzt der aus zwei Checkpoints besteht. Der Nutzer muss beide Checkpoints passieren und anschließend auf einem definierten Punkt landen. Mit einer Stoppuhr wird die benötigte Zeit erfasst, die als ein Indikator in einem direkten Vergleich der beiden Steuerungen dient.

IMPLEMENTIERUNG

Für unseren Prototyp haben wir Processing 2.2.1 verwendet. Damit auf den Leap Motion Sensor zugegriffen werden kann, wurde die Library „Leap Motion for Processing“ verwendet. Dieses Paket enthält allen nötigen Code, um eine Anwendung



3. Abbildung der Hürden

für Leap Motion zu schreiben. Ebenso wurde die Library „OBJLoader“ hinzugefügt, die es ermöglicht, 3D Objekte im *.obj Format zu importieren. Das Grundkonzept bei der Implementierung sieht wie folgendes aus. So wird in der draw() Methode, die jeden Frame aufgerufen wird, jedesmal auch eine Verschiebung des Copter Objektes in alle drei Achsenrichtungen durchgeführt. Wie groß diese Verschiebung ist, hängt von den Parametern der Hand ab. Alle Verschiebungen des 3D Copters werden durch Vektoren beeinflusst. Auf eine zusätzliche Simulation von Gravitation wurde verzichtet, da moderne GPS unterstützte Quadcopter ihre Position in der Luft, auch ohne Steuerung vom Benutzer, halten können. Für die Bewegung nach Links und Rechts bei „Variante A: Steuerung per Neigung“ bietet die importierte Library schon eine vorgefertigte Methode hand.getPitch(). Der Rückgabewert eignet sich direkt als Geschwindigkeitsindikator. Nähert sich die Neigung der Hand gegen 0, so ist auch die Bewegungsgeschwindigkeit nach links oder rechts gegen 0. Wird die Hand um z.B. um 40° geneigt, kann daraus eine Geschwindigkeit für den Copter abgeleitet werden. Die Steuerung für Vorwärts und Rückwärts verhält sich analog, jedoch mit der Funktion hand.getRoll(). Bei beiden Steuerungsvarianten reagiert die Drone sehr sensibel auf die Bewegungen für Vorwärts, Rückgabe, Links und Rechts. Dies hat den Vorteil, dass selbst feine Bewegungsmanöver zielgenau ausgeführt werden können. Für das Verändern der Höhe verhält es sich etwas anders. Hier war es gewünscht, den Copter absolut auf einer Höhe halten zu können, ohne dass kleinste

Handbewegungen den Copter zum ständigen steigen oder sinken veranlassen. Ein Schwellenwert von 0.25f ergab ein guten Kompromiss zwischen direktem Feedback und Stillhalten der Drone. Die Hand kann nun wenige Zentimeter hoch oder runterbewegt werden, ohne dass die Drone den Bewegungen der Hand folgt.

EVALUATION

Insgesamt wurden 5 Personen getestet. Abwechselnd fing ein Proband mit der Steuerung A, und Steuerung B an, um dadurch zu verhindern dass die Erkenntnisse aus dem ersten Durchgang das Ergebnis beeinflussen. Nach dem Test wurden jedem Proband folgende Fragen gestellt.

WAS HAT IHNEN AN DER STEUERUNG A GEFALLEN / NICHT GEFALLEN?

GEFALLEN:

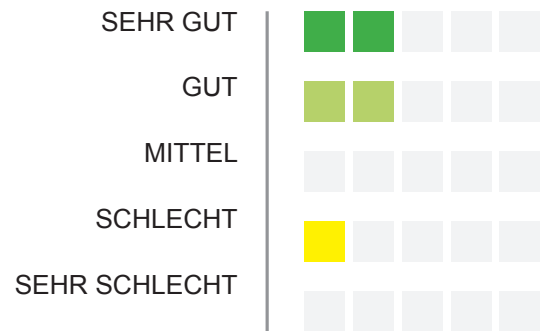
1. „Kleiner Bewegungsraum“
2. „Erinnert an Steuerung von Helikoptern in Videospielen. Also recht intuitiv,,
3. „Intuitiv, hat schnell und zuverlässig reagiert,,

NICHT GEFALLEN:

1. „Man muss stärker auf die Handneigung achten um die Drone stabil zu halten“
2. „Fehleranfällig, Leap Motion konnte Bewegung/Neigung

nicht immer erkennen, man wusste nicht wie weit man seine Hand neigen muss,,

3.“Daumen verkrampft,,



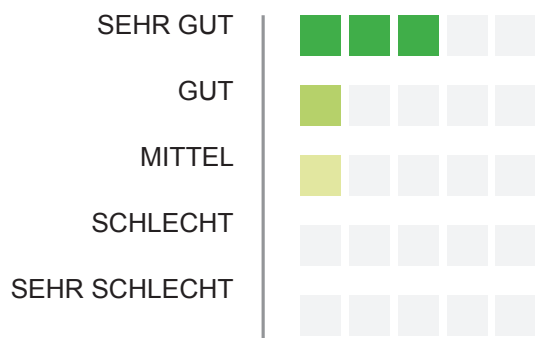
WAS HAT IHNEN AN DER STEUERUNG B GEFALLEN / NICHT GEFALLEN?

GEFALLEN:

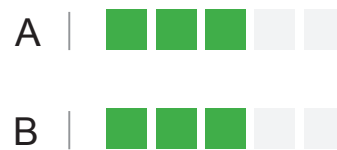
1. „War einfach zu steuern, Geschwindigkeit konnte besser eingeschätzt werden“
2. „Einfach, schneller Wechsel zwischen Positionen der Hand möglich. Schnelle Reaktionszeit. „
3. „Controls were more natural and intuitive,,
- 4.“Hand kann statisch gehalten werden, Bewegung sann analog zum „Keyboard“.,,

NICHT GEFALLEN:

1. „Größerer Bewegungsraum“
2. „Hat nicht ganz gut funktioniert, gefühlt.,,



KÖNNEN SIE SICH VORSTELLEN, DAMIT EINE ECHTE DRONE ZU STEuern? MIT WELCHER STEUERUNG?



WELCHE STEUERUNG WÜRDEN SIE BEVorzUGEN UND WIESO?



WEIL:

1. „Hat sich intuitiver und realer angefühlt, es war weniger Bewegung notwendig“



WEIL:

1. „Weil einfacher, leichter verständlich, Gesten werden gut erkannt“
2. „A wirkt erst intuitiver aber B scheint ein bisschen genauer,“
3. „More natural therefore easier to navigate,“
4. „Schnell zu lernen, analog zu Keyboard,“

GAB ES SCHWIERIGKEITEN BEIM STEuern ? WENN JA, WELCHE?

1. „Mein Daumen hat sich verkrampft“
2. „Die Ansicht bleibt und folgt nicht der Drone, daher ist die Entfernungsschätzung erschwert,“
3. „Geschwindigkeit einschätzen, Höhe beim Start nicht erkennbar,“
4. „Initiale Höhe der Hand war unklar (wie hoch muss die Hand zu Beginn sein?),“

WELCHE VERBESSERUNGsvORSCHLÄGE HABEN SIE?

1. „Konfiguration der Höhe beim Start bzw. beim Start anzeigen wann die Hand zu tief ist.“
2. „Pfeile für Richtung der aktuellen Geste anzeigen,“
3. „Kameraverfolgung der Drone beim fliegen,“

AUFFÄLLIGKEITEN BEI DER STUDIE

Abhängig davon mit welcher Steuerung die Probanden angefangen haben, versuchten sie die selben Gesten bei der zweiten Steuerung zu benutzen, obwohl diese keine Funktion mehr hatten. Wenn die Drone aus versehen aus dem Sichtbereich verschwand, waren die Probanden irritiert und waren nicht in der Lage das Objekt wieder in den sichtbaren Bereich zu navigieren. Mit der Variante B absolvierten die Probanden den Parkour deutlich schneller.

Die durchschnittliche Zeit die mit der Steuerung A benötigt wurde, beträgt: **27,4 SEKUNDEN**

Die durchschnittliche Zeit die mit der Steuerung B benötigt wurde, beträgt: **14 SEKUNDEN**

AUSBLICK

Nach der Userstudie können die gesammelten Erkenntnisse umgesetzt werden. Zusätzlich kann die Steuerung B, die von den getesteten Personen bevorzugt wurde, durch weitere Gesten und Funktionen erweitert werden. Nach einer Optimierung der Steuerung wäre das Ziel die Gesten mit einer echten Drone zu verbinden, und die Steuerungen zu testen.

FAZIT

Insgesamt haben beide Steuerungen gut funktioniert, die Tendenz geht zu einer absoluten Steuerung, ohne die Handfläche anwinkeln zu müssen. Der Nachteil allerdings ist dass man keinen sichtbaren Bereich hat in dem man die Hand bewegen kann, daher kann es schnell passieren dass die Hand nicht mehr vom Sensor aufgezeichnet wird. Zusätzlich bräuchte der Benutzer ein Feedback zu der aktuellen Geste, dadurch können orientierungslose Steuerungen des Objekts verhindert werden.